

# Pyrometer Interface

## *Serial port support*

Fri, May 16, 2003

A commercial device, the Mikron Infrared Pyrometer, is to be supported via RS-232 connection to an IRM in the Wideband lab. It measures temperature about twice a second and delivers it via RS-232 at 2400 baud, using 8 bits, 1 start, and 2 stops. This note describes the protocol used by the instrument and the support for same via a local application.

The output sent by the device consists of 9 characters, where the first one is a comma. The other 8 are as follows:

<i>Char</i>	<i>Meaning</i>
1	Sign, either a space or a minus sign
2	Most significant digit
3	Second digit
4	Third digit
5	Decimal point, if temperature below 1000, else space
6	Fourth digit, tenths if decimal point above, else units digit if above 1000
7	F or C, according to how the device is configured manually
8	space, or B to indicate Battery Low status.

Examples of digits 2–6, to illustrate the meaning of the decimal point are as follows:

<i>Chars 2–6</i>	<i>Temperature, C</i>
135.6	135.6
135 6	1356.

In addition to these normal readings, there are a few special sequences for alarm conditions:

<i>Chars 1–8</i>	<i>Meaning</i>
LLLLLLLLL	Low Temp Alarm
HHHHHHHH	High Temp Alarm
DDDDDDDD	Invalid Data Alarm (from Pyrometer internal processor)

Only a single reading is produced by this device, but several status indications are given:

Low Temp  
High Temp  
Invalid Data  
Battery Low  
Temperature Units (C or F)  
No data at all, or unrecognizable protocol

The first three conditions, plus the last one, can be accompanied by a bogus temperature value. The other two, Battery and Units, are status that can accompany valid numeric temperature values.

Since there is no CR, LF separating the successive temperature values, the serial input must be handled by monitoring the contents of the `SERIO` (Serial Input Queue). The `IN` offset of the queue header shows where the next character is to be placed. The `OUT1` is normally advanced only when a CR is encountered. But this device doesn't emit any, so a CR is inserted from time to time, every 251 characters. To process this data protocol, CR characters should be ignored. To prevent the `SERIO` from timing out, advance `OUT1` and `OUT2` to keep up with `IN`.

Suppose we assume the use of Centigrade units. A full scale value of 3276.8 allows for a range of

–3276.8 to +3276.7 degrees C. This permits a resolution of tenths of a degree. If units of F are indicated by the protocol, we can make the translation into C for storage. Even fancier, one could check the engineering units text used for F or C or DEGF or DEGC, but that may be overkill for this application.

Whenever a new value is detected, update the data pool value. If too much time passes without seeing a new value, then change the value to a bogus value to forestall misleading stale values. Status bits can be used to indicate the specific problem. An additional channel can hold an error number that can show the specific problem encountered when interpreting the protocol. A numeric value of –999.9 can be used when there is no recognizable numeric value.

Error values to indicate specific problems:

<i>Error</i>	<i>Meaning</i>
0	No errors, valid numeric protocol received
1	No data, no serial input data within timeout parameter
2	No comma as first character
3	Invalid sign character
4	Invalid decimal digit character
5	Invalid decimal point character
6	Invalid Units character
7	Invalid Battery Low character
8	n.u.
9	n.u.
10	Invalid Data (DDDDDDDD)
11	Low Alarm (LLLLLLLL)
12	High Alarm (HHHHHHHH)

Status bit indications, collected into one status byte:

<i>Bit#</i>	<i>Meaning</i>
7	No data, 1=None
6	Bad data in protocol, 1=Bad
5	Invalid Data, 1=Invalid
4	Low Alarm, 1=Low
3	High Alarm, 1=High
2	Units, 0=C, 1=F
1	Battery Low, 1=Low
0	n.u.

LA Parameters:

ENABLE	B	Enable Bit#
TEMP	C	Temperature Channel#
ERROR	C	Error Channel#
STATUS	B	Status base Bit#
TIMEOUT		Time-out in seconds

Diagnostics included in LA static memory block:

<i>Offset</i>	<i>Size</i>	<i>Diagnostic</i>
0x010	8	Time of last OK temperature reading
0x030	9	Input buffer
0x040	32	Error histogram
0x060	32	Error circular buffer
0x080	512	Circular buffer of temperature, error, status, and time-of-day.
0x280	256	Circular buffer of error-causing strings, with time-of-day.